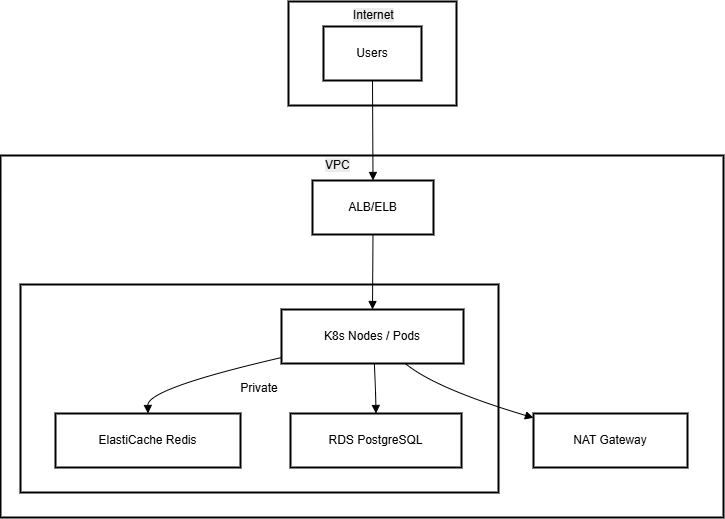
**Low‑Level Technical Specification: Infrastructure & DevOps**

Covers containerization, orchestration, CI/CD, environment segregation, IaC, monitoring, and disaster recovery, with Mermaid diagrams.

## **1. Environment Topology**

* **Cloud Provider**: AWS (preferred) or Azure/GCP
* **Regions**: Primary region (e.g., us-east-1) and DR region (e.g., us-west-2)
* **Accounts/Projects**: Separate AWS accounts for dev, staging, and prod
* **VPC Design**:  
  + Public subnets: Load balancers, NAT gateways
  + Private subnets: Application nodes, databases, Redis
  + Isolated subnets: Bastion host, management
* **Network ACLs & Security Groups**:  
  + LB SG: allow TCP 80/443 from internet
  + App SG: allow from LB SG, deny all else
  + DB SG: allow from App SG on port 5432



## **2. Containerization & Registry**

* **Docker**: multi‑stage Dockerfiles for backend, frontend, AI service
* **Base Images**: node:18-alpine, python:3.10-slim
* **Security Scanning**: Trivy in CI pipeline
* **Registry**: AWS ECR (private) or Docker Hub (private org)
* **Image Tagging**: registry/app:service-{commitSha}, promote tags service-latest, service-v{major.minor}
* **Immutable Images**: no latest in prod; use SHA tags

## **3. Orchestration (Kubernetes)**

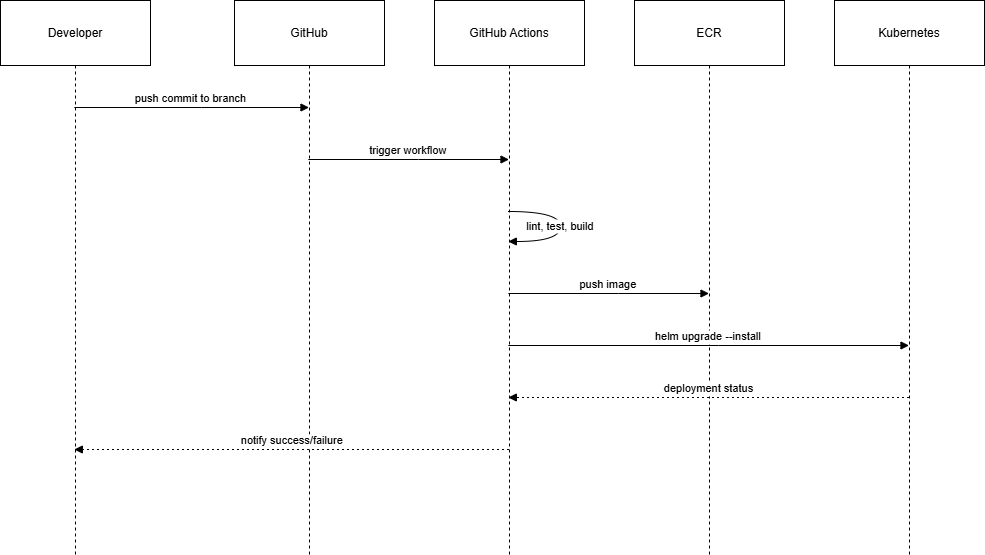
* **K8s Flavor**: EKS (AWS), AKS (Azure) or GKE
* **Cluster Sizing**:  
  + Min nodes: 3 x t3.medium (dev), 5 x m5.large (prod)
  + Node Pools: separate pools for “standard” and “GPU/AI” workloads
* **Namespaces**: dev, staging, prod
* **Workloads**:  
  + Deployments for API, frontend, AI service
  + StatefulSets for Redis (if self‑managed) and PostgreSQL (if self‑managed)
* **Config & Secrets**:  
  + ConfigMaps for non-sensitive configs
  + Kubernetes Secrets for DB creds, OAuth keys
* **Ingress Controller**: AWS ALB Ingress Controller or NGINX Ingress
* **Autoscaling**:  
  + HPA based on CPU (target 60%) and custom metric (e.g., request latency)
  + Cluster autoscaler
* **Service Mesh (Optional)**: Istio or Linkerd for mTLS and traffic management

## **4. Infrastructure as Code (Terraform)**

* **Modules**:  
  + VPC and networking
  + EKS cluster and node groups
  + RDS and ElastiCache
  + IAM roles and policies
* **State Management**: S3 backend with DynamoDB lock table
* **Workspace Segregation**: Terraform workspaces for dev, staging, prod
* **PR Workflow**:  
  + terraform plan on PR
  + terraform apply on merge to main (for prod) or dedicated branches (for dev/staging)

## **5. CI/CD Pipeline**

* **Platform**: GitHub Actions (or GitLab CI)
* **Stages**:  
  1. **Checkout & Lint**: ESLint, Stylelint, terraform fmt
  2. **Unit & Integration Tests**: Jest, pytest, supertest, socket.io-client
  3. **Build & Scan**: Docker build + Trivy scan
  4. **Push Images**: to ECR with commit SHA tag
  5. **Deploy**:  
     + dev: auto‑deploy on any push to dev branch
     + staging: deploy on PR merge to main branch
     + prod: manual approval gate before deploy
* **Rollback**: Helm rollbacks or Kubernetes kubectl rollout undo



## **6. Monitoring & Logging**

* **Metrics**: Prometheus scraping:  
  + API request rates, latencies, errors
  + WebSocket connections and events
  + AI service inference times
* **Dashboards**: Grafana:  
  + Overview: cluster health, service health
  + Detailed: per‑service metrics
* **Logging**: EFK stack:  
  + Fluentd / Promtail → Elasticsearch / Loki
  + Dashboards in Kibana / Grafana
* **Alerting**: Prometheus Alertmanager:  
  + High error rates (>5% 5m)
  + CPU/Memory >80% for >5m
  + Pod restarts >3 in 10m

## **7. Secrets Management**

* **Kubernetes Secrets** for small teams
* **Vault (Optional)** for dynamic DB creds
* **Rotation**: vault-issued secrets rotated every 30 days, Kubernetes secrets updated via CI/CD

## **8. Backup & Disaster Recovery**

* **PostgreSQL Backups**:  
  + Automated snapshots daily
  + WAL archiving for PITR
* **Redis Backups**:  
  + RDB snapshots hourly
* **Cluster State**:  
  + Terraform state in S3 with versioning
* **DR Drills**:  
  + Quarterly restore tests to DR region
  + Documented runbooks for recovery

## **9. Access & Governance**

* **IAM Policies**:  
  + Least privilege for service accounts
  + Role separation: infra admins vs devs
* **SAML SSO**: integrate AWS Console, Keycloak Admin UI
* **Audit Trails**:  
  + AWS CloudTrail for API changes
  + Kubernetes audit logs

*End of Infrastructure & DevOps Specification*